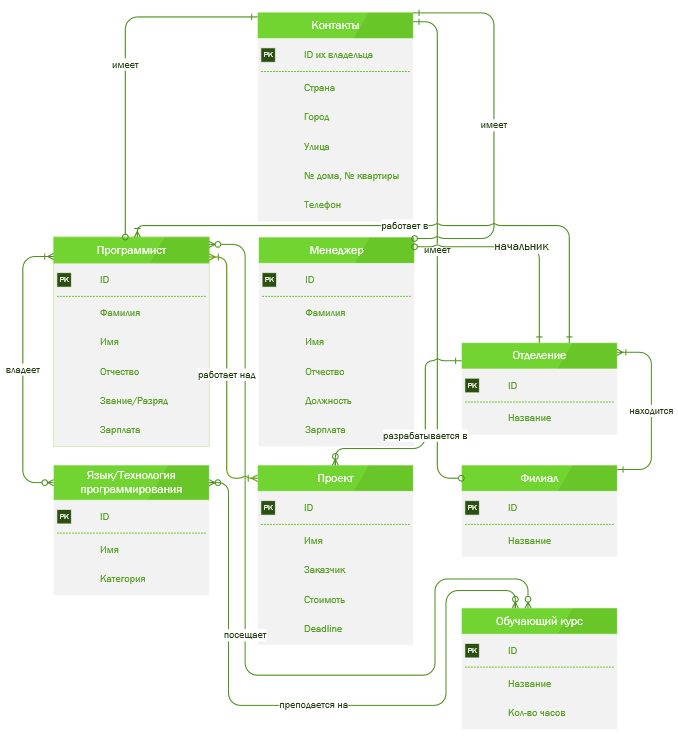
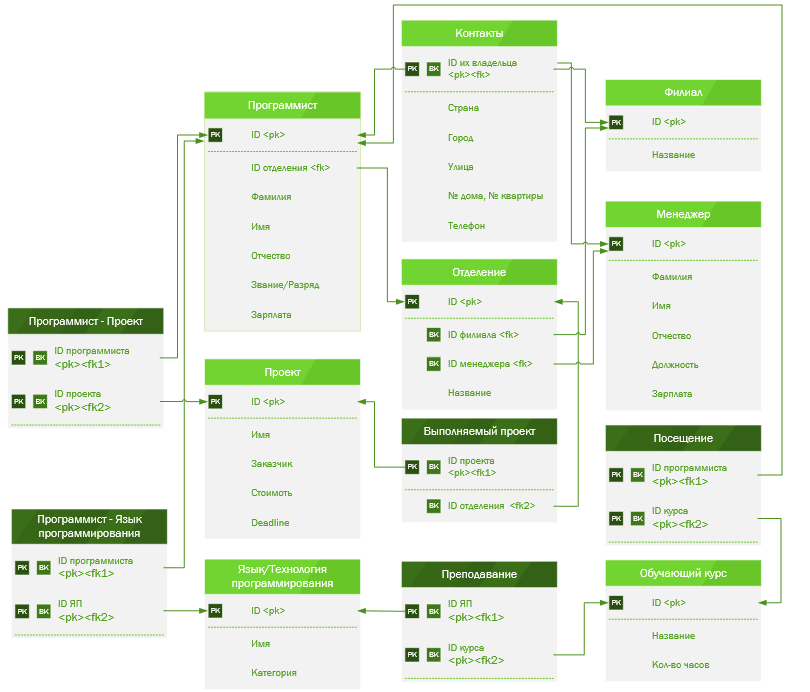
Задание №3

**Физическая схема реляционной базы данных**

*Подготовил Желтоухов Максим, 4 группа*



*ER-диаграмма предметной области «IT-Компания»*



*Физическая схема реляционной базы данных, преобразованная из ER-диаграммы*

**Обозначения:**

* <pk> - первичный ключ отношения
* <fk>/<fk1>/<fk2> - внешние ключи отношения

**Пояснения:**

**Связь один-к-одному обязательная с одной стороны:**

* На ER-диаграмме сущность **«Контакты»** связана с сущностями **«Программист», «Менеджер», «Филиал»** связью 1:1, обязательной со стороны сущности «Контакты». В реляционной схеме атрибут «ID их владельца» в отношении «Контакты» является его первичным ключом и, одновременно, внешним ключом, ссылающимся на первичные ключи «ID» отношений «Программист», «Менеджер», «Филиал». Поэтому такая связь реализуется с помощью ссылки внешнего ключа к родительскому первичному ключу.
* На ER-диаграмме сущность **«Менеджер»** связана с сущностью **«Отделение»** связью 1:1, обязательной со стороны сущности «Отделение». В отношение, соответствующее сущности со стороны обязательного класса принадлежности («Отделение»), вводится новый атрибут «ID менеджера», определенный на том же домене, что атрибут «ID» в отношении «Менеджеры». Этот атрибут связан с первичным ключом «ID» (отделения) взаимной функциональной зависимостью. При этом «ID менеджера» становится внешним ключом, ссылающимся на первичный ключ «ID» (менеджера).

**Связь многие-к-одному обязательная с обеих сторон:**

На ER-диаграмме представлены две пары сущностей «Программист», «Отделение» и «Отделение», «Филиал», связанные связью один-ко-многим с обязательным классом принадлежности для обеих сторон связи. Такая связь реализуется в реляционной схеме с помощью функциональной зависимости, поэтому были проведены следующие изменения:

* **«Программист»** и **«Отделение»**:

В отношение, соответствующее сущности с множественной связностью («Программист»), вводится атрибут «ID отделения», который становится внешним ключом, ссылающимся на одноименный первичный ключ отношения Владельцы автомобилей «Отделение».

* **«Отделение»** и **«Филиал»**:

В отношение, соответствующее сущности с множественной связностью («Отделение»), вводится атрибут «ID филиала», который становится внешним ключом, ссылающимся на одноименный первичный ключ отношения Владельцы автомобилей «Филиал».

*Примечание: реляционная модель не в состоянии реализовать обязательность класса принадлежности связи один-ко-многим со стороны сущности с одинарной связностью. Поэтому, схема не препятствует вводу в отношения «Отделение» и «Филиал» новых записей, с которыми не связаны ни один программист и ни одно отделение соответственно.*

**Связь многие-ко-многим обязательная с обеих сторон:**

* На ER-диаграмме представлены сущности **«Программист»** и **«Проект»**, связанные друг с другом связью многие-ко-многим обязательной с обеих сторон. Эта ER-диаграмма преобразуется в реляционную схему, содержащую три отношения. Это отношение «Программист» и отношение «Проект», которые соответствуют сущностям исходной ER-модели, и также третье отношение «Программист-проект», составной первичный ключ которого {ID программиста, ID проекта} реализует обязательную с обеих сторон связь многие-ко-многим. Каждый из этих атрибутов одновременно является внешним ключом.

*Примечание: в полученной реляционной схеме класс принадлежности связи со стороны сущностей оказался необязательным, в отличие от исходной ER-диаграммы, так как реляционная схема не позволяет реализовать обязательный класс принадлежности для связи многие-ко-многим m:n.*

**Связь многие-к-одному обязательная с одной стороны:**

* На ER-диаграмме представлены сущности **«Проект»** и **«Отделение»**, связанные друг с другом связью многие-к-одному обязательной со стороны сущности «Отделение». Эта ER-диаграмма преобразуется в реляционную схему, содержащую три отношения. Это отношение «Проект» и отношение «Отделение», которые соответствуют сущностям исходной ER-модели, и также третье отношение «Выполняемый проект» с атрибутом «ID проекта» в качестве первичного ключа. Добавленное отношение реализует связь один-ко-многим между сущностями «Проект» и «Отделение». Внешний ключ этого отношения – атрибут «ID проекта», ссылающийся на первичный ключ отношения «Проекты», реализует необходимый для этого примера необязательный класс принадлежности связи с сущностью «Проект».

*Примечание: второй внешний ключ отношения «Выполняемое задание» своей ссылкой на первичный ключ отношения «Отделение» реализует необязательный класс принадлежности связи со стороны сущности «Отделение», что не соответствует исходной ER-диаграмме, так как реляционная модель не позволяет реализовать обязательность класса принадлежности со стороны сущности с одинарной связностью.*

**Связь многие-ко-многим обязательная с одной стороны:**

* На ER-диаграмме представлены сущности **«Программист»** и **«Языки программирования»**, связанные друг с другом связью многие-ко-многим обязательной только со стороны сущности «Программист». Эта ER-диаграмма преобразуется в реляционную схему, содержащую три отношения. Это отношения «Программист» и «Языки программирования», которые соответствуют сущностям исходной ER-модели, и третье отношение «Программист – язык программирования», составной первичный ключ которого {ID программиста, ID ЯП} реализует обязательную с обеих сторон связь многие-ко-многим. Каждый из этих атрибутов одновременно является внешним ключом, ссылающимся на первичный ключ соответствующего отношения, реализуя необязательный класс принадлежности к сущностям, порождающим эти отношения.

*Примечание: в полученной реляционной схеме класс принадлежности связи с обеих сторон оказался необязательным, в отличие от исходной ER-диаграммы, так как реляционная схема не позволяет реализовать обязательный класс принадлежности для сторон связи m:n.*

**Связь многие-ко-многим необязательная с обеих сторон:**

На ER-диаграмме представлены две пары сущностей «Программист», «Обучающий курс» и «Язык программирования», «Обучающий курс», связанные связью многие-ко-многим с обязательным классом принадлежности для обеих сторон связи. Такая ER-диаграмма преобразуется в реляционную схему, содержащую три отношения:

* **«Программист»** и **«Обучающий курс»**

Получаем отношение «Программист» и отношение «Обучающий курс», соответствующие сущностям исходной ER-модели, и третье отношение «Посещение», составной первичный ключ которого {ID программиста, ID курса} реализует обязательную с обеих сторон связь многие-ко-многим. Каждый из этих атрибутов одновременно является внешним ключом, реализующим необязательный класс принадлежности со стороны родительского отношения.

* **«Язык программирования»** и **«Обучающий курс»**

Получаем отношение «Язык программирования» и отношение «Обучающий курс», соответствующие сущностям исходной ER-модели, и третье отношение «Преподавание», составной первичный ключ которого {ID ЯП, ID курса} реализует обязательную с обеих сторон связь многие-ко-многим. Каждый из этих атрибутов одновременно является внешним ключом, реализующим необязательный класс принадлежности со стороны родительского отношения.